Summary - Back Pressure

Name	backpressure
Worker Type	Application
Version	v1.3
Release Date	9/2018
Component Library	ocpi.core
Workers	backpressure.hdl, backpressure.rcc
Tested Platforms	isim, xsim, modelsim, xilinx13_3, centos6, centos7, alst4, ml605, ZedBoard(PL), Matchstiq-Z1(PL)

Functionality

The Back Pressure component provides the ability to emulate 'back pressure' that is present in a system. It is primarily used during the development of an HDL worker, specifically during unit test simulations. The *backpressure* worker is built into a worker's unit test HDL assembly and is used to force 'back pressure' during the execution of application to exercise the worker's ability to correctly handle 'back pressure'.

This worker does not manipulate the data, but simply passes it through. Validation of this worker requires passing a known input data pattern through the worker under its various modes and comparing the input and output files to verify that the data is unchanged. Since validation of the output is performed simply by comparing to the input, any non-zero input data would be sufficient.

Worker Implementation Details

backpressure.hdl

The Back Pressure worker does not define input/output protocols explicitly. Since the input is simply bits, the input protocol is irrelevant and defined by the component feeding the Back Pressure, such as the File Reader. This worker only applies 'back pressure' to that worker which is upstream within the application.

backpressure.rcc

The RCC version of this component is just a placeholder to fulfill the requirements of unit test framework. It passes through data without change and shouldn't be included in normal applications, as it provides no real functionality.

Theory

Back pressure within a system is a common occurrence that can be a result of resource loading issues or passing data between containers. Workers must be designed to handle system back pressure without data loss.

Block Diagrams

Top level

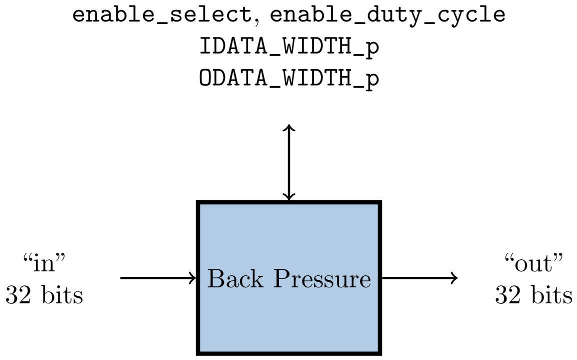

Figure 1: Top Level Block Diagram

State Machine

N/A

Source Dependencies

backpressure.hdl

- projects/core/components/backpressure.hdl/backpressure.vhd

backpressure.rcc

- projects/core/components/backpressure.rcc/backpressure.cc

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
`enable_select`	bool	-	-	Readable, Writable	Standard	False	Select back pressure scheme to control 'take' from upstream worker. True = uses lfsr-15 or False = uses configurable duty cycle
`enable_duty_cycle`	ushort	-	-	Readable, Writable	Standard	1	Set 'take' duty cycle: 1 = constant, 2 = toggle, 3 = 1/on 2/off, 4 = 1/on 3/off, etc.

Worker Properties

backpressure.hdl

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
`IDATA_WIDTH_p`	ulong	-	-	Readable, Parameter	8/16/32/64	32	Input port data width
`ODATA_WIDTH_p`	ulong	-	-	Readable, Parameter	8/16/32/64	32	Output port data width

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
in	False	-	False	-	32 bits
out	True	-	False	-	32 bits

Worker Interfaces

backpressure.hdl

Type	Name	DataWidth	Advanced	Usage
StreamInterface	in	`IDATA_WIDTH_p`	-	Size defined by `IDATA_WIDTH_p`
StreamInterface	out	`ODATA_WIDTH_p`	-	Sample size defined by `ODATA_WIDTH_p`

Control Timing and Signals

backpressure.hdl

This worker implementation uses the clock from the Control Plane and standard Control Plane signals.

Worker Configuration Parameters

backpressure.hdl

Table 1: Table of Worker Configurations for worker: backpressure

Configuration
0

Performance and Resource Utilization

backpressure.hdl

Table 2: Resource Utilization Table for worker ”backpressure”

Configuration	OCPI Target	Tool	Version	Device	Registers (Typ)	LUTs (Typ)	Fmax (MHz) (Typ)	Memory/Special Functions
0	zynq_ise	ISE	14.7	7z010clg400-3	334	485	342.874	N/A
0	zynq	Vivado	2017.1	xc7z020clg400-3	326	318	N/A	N/A
0	stratix4	Quartus	17.1.0	N/A	328	212	N/A	N/A
0	virtex6	ISE	14.7	6vcx75tff484-2	334	564	313.65	N/A

Test and Verification

This component is tested via the unit test automation feature of the framework. The component's .test/ contains XML files that describe the combinations of tests.

Fundamentally, there are two test cases that are employed to verify the Back Pressure component:

1. enable_select = True: The most significant bit of the lfsr-15 drives the 'take' signal of the input port.
2. enable_select = False: The enable_duty_cycle setting controls the duty cycle of the 'take' signal of the input port.

In all test cases, the data is simply passed through the worker and the tests are determined to be successful by comparing the input and output files. Due to its simplicity, and usage in other unit tests, a binary data file is generated containing complex signed 16-bit samples with a tone at a configurable center frequency and sample frequency. Plotting of the I/O data is available, via View=1, if desired.